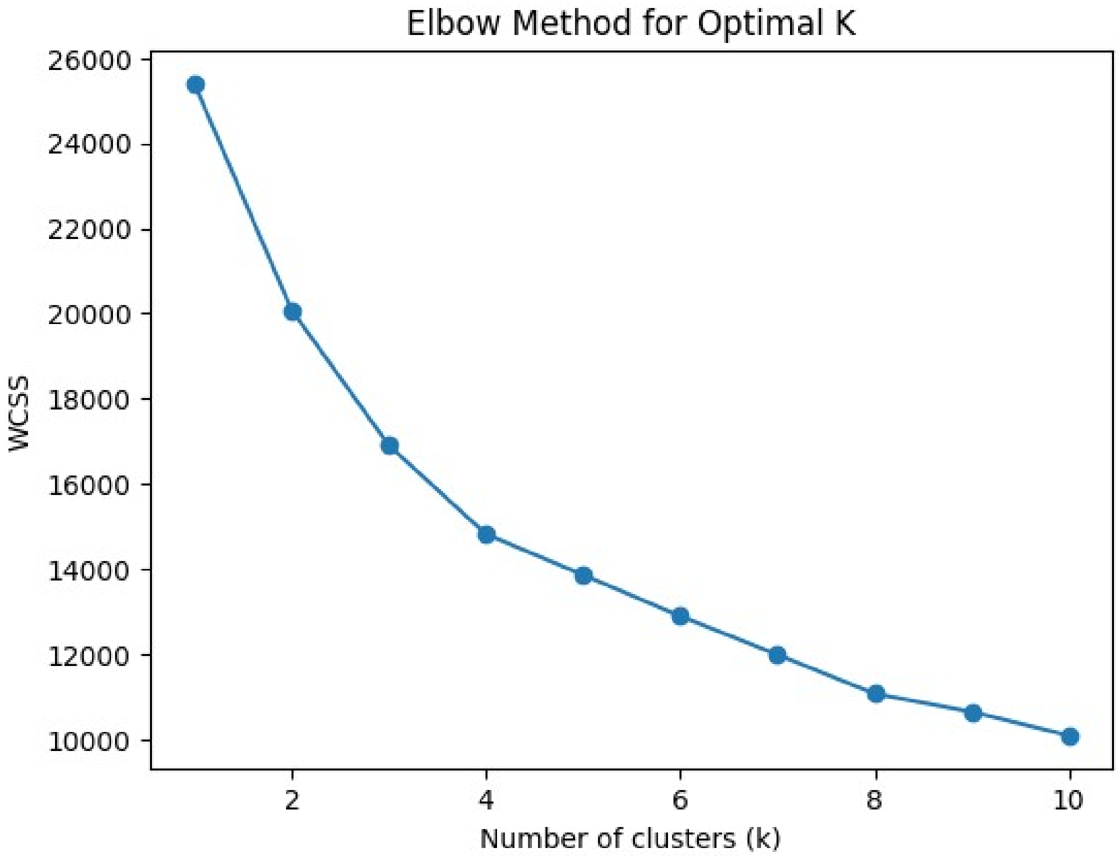
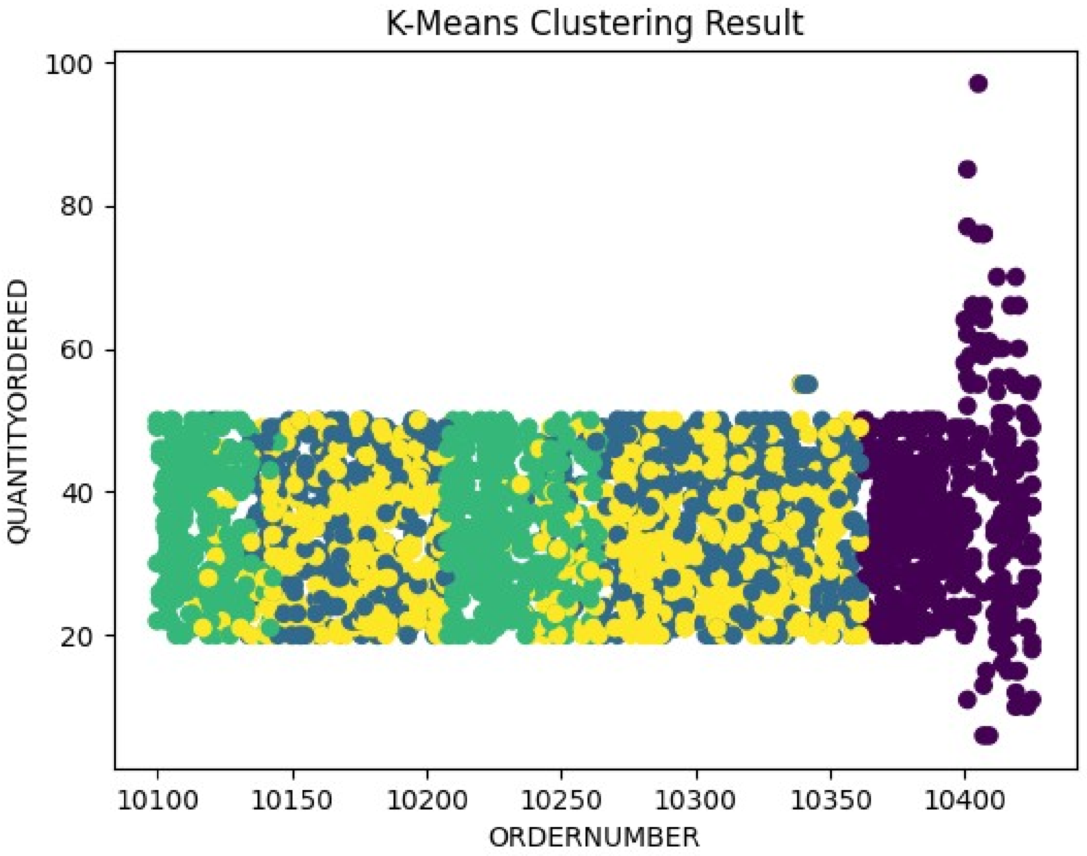
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from scipy.cluster.hierarchy import linkage, dendrogram, fcluster

# Load the dataset
data = pd.read_csv("/content/sales_data_sample.csv",
encoding='latin1')
data.head()
```

```
{"type":"dataframe","variable_name":"data"}
```

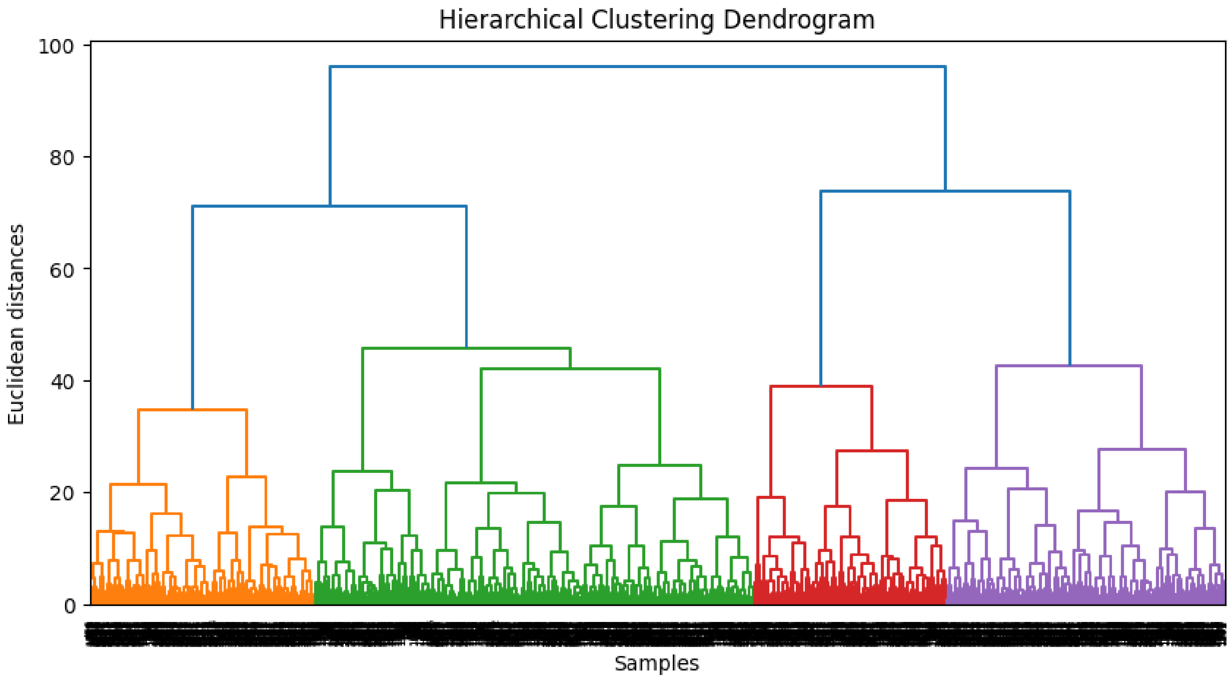
```python
# Select numeric columns for clustering
num_data = data.select_dtypes(include=['float64', 'int64']).copy()

# Handle missing values if any
num_data = num_data.dropna()

# Standardize the data
scaler = StandardScaler()
scaled_data = scaler.fit_transform(num_data)

wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, random_state=42)
    kmeans.fit(scaled_data)
    wcss.append(kmeans.inertia_)

# Plot the elbow curve
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Elbow Method for Optimal K')
plt.xlabel('Number of clusters (k)')
plt.ylabel('WCSS')
plt.show()
```



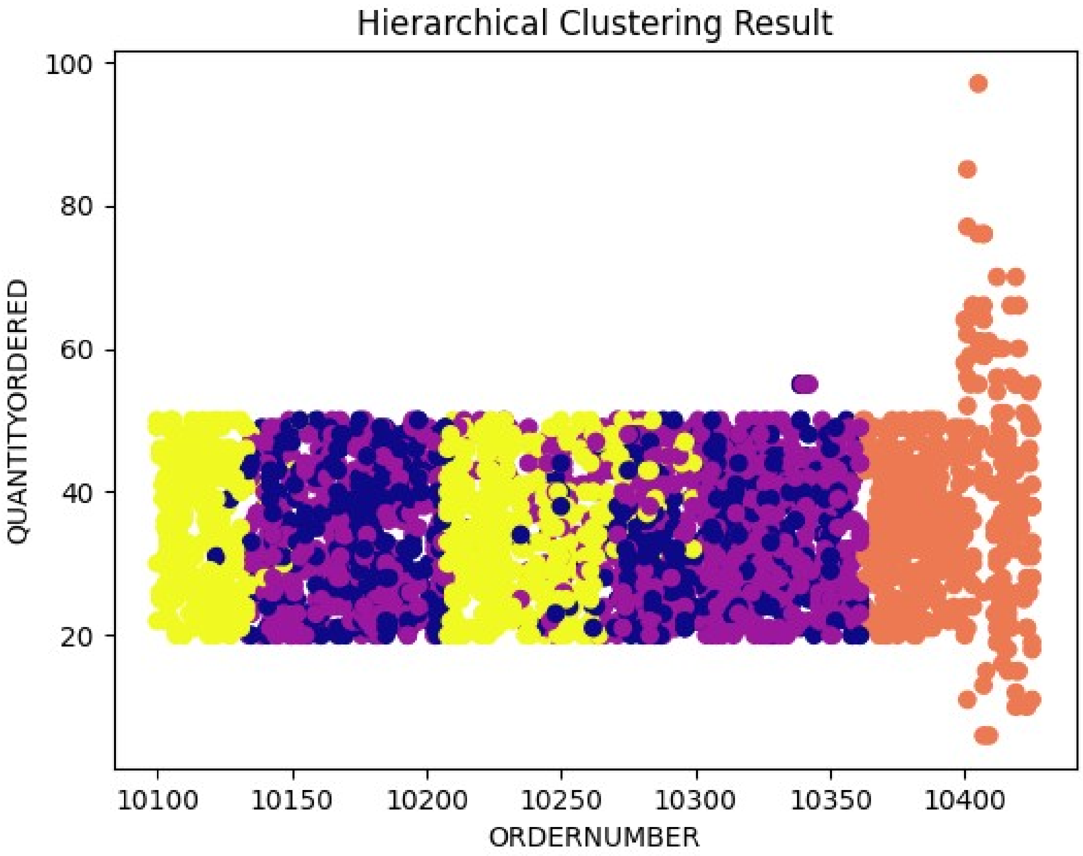
```
# Choose k based on elbow result (for example, k=4)
kmeans = KMeans(n_clusters=4, random_state=42)
kmeans_labels = kmeans.fit_predict(scaled_data)

# Add cluster labels to dataset
num_data['Cluster'] = kmeans_labels

# Visualize clusters using two features
plt.scatter(num_data.iloc[:, 0], num_data.iloc[:, 1], c=kmeans_labels,
cmap='viridis')
plt.title('K-Means Clustering Result')
plt.xlabel(num_data.columns[0])
plt.ylabel(num_data.columns[1])
plt.show()
```



```
# Perform hierarchical clustering using Ward’s method
Z = linkage(scaled_data, method='ward')

# Plot dendrogram
plt.figure(figsize=(10, 5))
dendrogram(Z)
plt.title('Hierarchical Clustering Dendrogram')
plt.xlabel('Samples')
plt.ylabel('Euclidean distances')
plt.show()
```



```python
# Suppose we choose 4 clusters based on the dendrogram
hier_labels = fcluster(Z, t=4, criterion='maxclust')
num_data['HCluster'] = hier_labels

# Visualize first two features
plt.scatter(num_data.iloc[:, 0], num_data.iloc[:, 1], c=hier_labels,
cmap='plasma')
plt.title('Hierarchical Clustering Result')
plt.xlabel(num_data.columns[0])
plt.ylabel(num_data.columns[1])
plt.show()
```


Hierarchical Clustering Result
100
80
60
40
20
QUANTITYORDERED
10100
10150
10200
10250
10300
10350
10400
ORDERNUMBER